

Number Magic II

num (> 9) be a positive integer. sum is defined as sum_of_digits_of_num. avg is defined as sum / digits_in_num and avg takes only integer values (for example $34 / 5$ is equal to 6, i.e. integer part of 6.8). Now digits of num are rearranged by taking avg as pivoting element. Rearrangement is done according the following method -

[step 1]

new_num = avg;

read digits of num

if (digit > avg)

place digit at rightmost place in new_num

else

place digit at leftmost place in new_num

[step 2]

new_num is again rearranged into two numbers, eve_number (**formed by taking digits of new_num at places 2, 4, 6, ...**) and odd_number (**formed by taking digits of new_num at places 1, 3, 5, ...**).

Remember that counting of digit place starts from leftmost digits as 1, 2, 3, 4...

[step 3]

if (eve_number has < 3 digits)

store the number

else

calculate avg and again start from process 1

if (odd_number has < 3 digits)

store the number

else

calculate avg and again start from step 1

Finally all desired numbers are stored and now we have to find two magical coefficients of num, named as alpha and beta.

[method digit_sum]

number = num

do {

number = sum of digits of number

} while (number \geq 10)

dig_sum = number

[alpha]

digit_sum(summation_of_stored_numbers)

[beta]

digit_sum(summation_of_digit_sum(stored_numbers))

we will say num is magic number if alpha, beta and $|\alpha - \beta|$ are digits of num.

Input

First line of input is t (< 101), total number of test cases. Each test case has n (< 501) as its first input and next n lines contains num ($< 10^{101}$).

Output

For each test case, write exactly n lines containing value of alpha and beta; and yes or no according to whether or not num is magic number.

Example

Input:

1
1
9874

Output:

5 5 no

Explanation

num = 9874

avg = 7 (28 / 4)

rearrangement of num generates new_num = 47798

thus eve_num = 79 and odd_num = 478

now eve_num < 100, so 79 is stored.

but odd_num >= 100 so repeat the full process.

Finally stored numbers are N[] = 79, 68 and 47

Make sure that you have to consider the total number of digits not the value of the number so you should not change the total number of digits of the number at any step